UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/802,198 | 03/16/2004 | Daniel P. Baumberger | P18511 | 3760 |

59796 7590 06/09/2009

INTEL CORPORATION
c/o CPA Global
P.O. BOX 52050
MINNEAPOLIS, MN 55402

| EXAMINER | |
|---|---|
| PATEL, JAY P | |
| ART UNIT | PAPER NUMBER |
| 2419 | |
| MAIL DATE | DELIVERY MODE |
| 06/09/2009 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| | | | |
|---|---|---|---|
| | **Application No.** | **Applicant(s)** | |
| ***Office Action Summary*** | 10/802,198 | BAUMBERGER, DANIEL P. | |
| | **Examiner** | **Art Unit** | |
| | JAY P. PATEL | 2419 | |

***-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --***

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

1)☒ Responsive to communication(s) filed on *22 April 2009*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

## Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) □ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.
4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) □ Notice of Informal Patent Application
6) □ Other: _____.

## DETAILED ACTION

The rejection under 35 USC § 101 presented in the previous office action has been withdrawn in view of the amendment filed on 4/22/2009 where the scope of claim 12 refers to a computer-readable storage media. It is noted that the storage media does not include electrical, optical, acoustical or other form of propagated signals (such as carrier waves, infrared signals and digital signals).

### *Claim Rejections - 35 USC § 103*

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bugnion et al. (US Patent 6075938) in view of Carrozza et al. (US Patent 6445685 B1) further in view of Wang (US Patent 6477612 B1) further in view of Paladini et al. (US Publication 2006/0123215 A1).

3. In regards to claim 1, Bugnion shows in figure 3, a memory management data structure consisting of two virtual machines with two examples of memory management. The second example shows the impact of a page migration action. Also evident from the figure itself, a virtual address is indexed to a physical address. The transparent migration requires that all mappings that point to that page be removed from all processors (unmapping a guest physical address from a host physical address in at

least one page table entry associated with buffers in a DMA table to create unmapped buffers) (see column 14, lines 19-30).

In further regards to claim 1, Bugnion fails to teach a demultiplexing operation where an incoming packet is placed into a buffer. Carrozza however teaches the above-mentioned limitation in figure 5 and 6 for memory allocation of data demultiplexing (see column 12, lines 56-57).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use the unmapping and mapping of buffers taught by Bugnion with the demultiplexing process taught by Carrozza. The motivation to do so would be to support a buffer that is shared by multiple virtual machines (see the conclusion paragraph in column 7 in Bugnion).

In further regards to claim 1, Bugnion and Carrozza fail to teach allocating unmapped buffers to the virtual machine to create a mapped buffer. Wang however, teaches the above-mentioned limitation. Wang teaches than an API function allocates to a process the physical memory pages that may be mapped and unmapped within any specially-allocated virtual address space region of the specified process (see column 7, lines 10-13).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the buffer allocation taught by Wang into the unmapping and mapping of buffers taught by Bugnion and the demultiplexing process taught by Carrozza. The motivation to do so would be to allow for fast mapping for a multiprocessor system (see column 2, lines 9-12).

In further regards to claim 1, Bugnion, Carrozza and Wang fail to teach the concept of unmapping a buffer space from a first virtual machine and the mapping the same space to a second virtual machine. Paladini however teaches the above-mentioned concept. Paladini teaches un-mapping a buffer from the virtual address space of the application (un mapping from a first virtual machine) and mapping it back later to a different virtual address (mapping it to a second virtual machine) different from the initial assigned address (see paragraph 58 on page 4).

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to incorporate the concept of unmapping and mapping a buffer from one machine to another as taught by Paladini into the teachings of Bugnion, Carrozza and Wang. The motivation to do so would be to use an updated memory manager.

In regards to claim 2, since Bugnion teaches that the transparent migration requires that all mappings that point to a page be removed form all processors (see column 14, lines 19-30), it also reads on clearing the contents of a physical page associated with the host physical address.

In regards to claim 3, the mechanism shown in figure 3 of Bugion, allows for a support of system-wide cache in memory that can be shared between all virtual machines (therefore, reading on a temporary association between a mapped and an unmapped buffer).

In regards to claim 4, since Bugnion teaches that the transparent migration requires that all mappings that point to a page be removed form all processors (see column 14, lines 19-30), it also reads on causing the VM to release the mapped buffer and unmapping the guest physical address from the host physical address.

In regards to claims 5 and 6, disco intercepts (injecting a signal and intercepting) all device accesses from the virtual machines and forwards them to the physical address (see column 14, lines 32-34 in Bugnion).

4. In regards to claim 7, Bugnion shows in figure 3, a memory management data structure consisting of two virtual machines (plurality of virtual machines) with two examples of memory management. The second example shows the impact of a page migration action. Also evident from the figure itself, a virtual address is indexed to a physical address. The transparent migration requires that all mappings that point to that page be removed from all processors (invalidating entries in at least one page table entry for direct memory access buffers to create unmapped buffers) (see column 14, lines 19-30).

In further regards to claim 7, Bugnion fails to teach a demultiplexing operation where an incoming packet is placed into an appropriate buffer. Carrozza however teaches the above-mentioned limitation in figures 5 and 6 for memory allocation of data demultiplexing (see column 12, lines 56-57).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use the unmapping and mapping of buffers taught by Bugnion with the demultiplexing process taught by Carrozza. The motivation to do so would be to support a buffer that is shared by multiple virtual machines (see the conclusion paragraph in column 7 in Bugnion).

In further regards to claim 7, Bugnion and Carrozza fail to teach allocating unmapped buffers to a proper virtual machine. Wang however, teaches the above-mentioned limitation. Wang teaches than an API function allocates to a process the physical memory pages that may be mapped and unmapped within any specially-allocated virtual address space region of the specified process (see column 7, lines 10-13).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the buffer allocation taught by Wang into the unmapping and mapping of buffers taught by Bugnion and the demultiplexing process taught by Carrozza. The motivation to do so would be to allow for fast mapping for a multiprocessor system (see column 2, lines 9-12).

In further regards to claim 7, Bugnion, Carrozza and Wang fail to teach the concept of unmapping a buffer space from a first virtual machine and the mapping the same space to a second virtual machine. Paladini however teaches the above-mentioned concept. Paladini teaches un-mapping a buffer from the virtual address space of the application (un mapping from a first virtual machine) and mapping it back later to a different virtual address (mapping it to a second virtual machine) different from the initial assigned address (see paragraph 58 on page 4).

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to incorporate the concept of unmapping and mapping a buffer from one machine to another as taught by Paladini into the teachings of Bugnion, Carrozza and Wang. The motivation to do so would be to use an updated memory manager.

In regards to claim 8, since Bugnion teaches that the transparent migration requires that all mappings that point to a page be removed form all processors (see column 14, lines 19-30), it also reads on invalidating entries in at least one page in a DMA table (also see column 14, lines 28-30 and figure 3).

5. In regards to claim 9, Bugnion shows in figure 3, a memory management data structure consisting of two virtual machines (plurality of virtual machines) with two examples of memory management. The second example shows the impact of a page migration action. Also evident from the figure itself, a virtual address is indexed to a physical address. The transparent migration requires that all mappings that point to that page be removed from all processors (invalidating entries in at least one page table entry for direct memory access buffers to create unmapped buffers) (see column 14, lines 19-30).

In further regards to claim 9, Bugnion fails to teach a demultiplexing operation where an incoming packet is placed into an appropriate buffer. Carrozza however teaches the above-mentioned limitation in figures 5 and 6 for memory allocation of data demultiplexing (see column 12, lines 56-57).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use the unmapping and mapping of buffers taught by Bugnion with the demultiplexing process taught by Carrozza. The motivation to do so would be to support a buffer that is shared by multiple virtual machines (see the conclusion paragraph in column 7 in Bugnion).

In further regards to claim 9, Bugnion and Carrozza fail to teach allocating unmapped buffers to a proper virtual machine. Wang however, teaches the above-mentioned limitation. Wang teaches than an API function allocates to a process the physical memory pages that may be mapped and unmapped within any specially-allocated virtual address space region of the specified process (see column 7, lines 10-13).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the buffer allocation taught by Wang into the unmapping and mapping of buffers taught by Bugnion and the demultiplexing process taught by Carrozza. The motivation to do so would be to allow for fast mapping for a multiprocessor system (see column 2, lines 9-12 in Wang).

In further regards to claim 9, Bugnion, Carrozza and Wang fail to teach the concept of unmapping a buffer space from a first virtual machine and the mapping the same space to a second virtual machine. Paladini however teaches the above-mentioned concept. Paladini teaches un-mapping a buffer from the virtual address space of the application (un mapping from a first virtual machine) and mapping it back later to a different virtual address (mapping it to a second virtual machine) different from the initial assigned address (see paragraph 58 on page 4).

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to incorporate the concept of unmapping and mapping a buffer from one machine to another as taught by Paladini into the teachings of Bugnion, Carrozza and Wang. The motivation to do so would be to use an updated memory manager.

In regards to claim 10, Bugnion in combination with Carrozza and Wang teaches all the limitations of parent claim 9. Carrozza shows a demultiplexing operation in figures 5 and 6. Bugnion also shows that two virtual machines are mapped to an address space; therefore, an interface card must be present.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the buffer allocation taught by Wang into the unmapping and mapping of buffers taught by Bugnion and the demultiplexing process taught by Carrozza. The motivation to do so would be to support a buffer that is shared by multiple virtual machines (see the conclusion paragraph in column 7 in Bugnion).

In regards to claim 11, since multiple virtual machines are used in Bugnion, a virtual machine manager must be present and coupled to the two virtual machines.

6. In regards to claim 12, Bugnion shows in figure 3, a memory management data structure consisting of two virtual machines with two examples of memory management. The second example shows the impact of a page migration action. Also evident from the figure itself, a virtual address is indexed to a physical address. The transparent migration requires that all mappings that point to that page be removed from all processors (unmapping a guest physical address from a host physical address in at

least one page table entry associated with buffers in a DMA table to create unmapped buffers) (see column 14, lines 19-30).

In further regards to claim 12, Bugnion fails to teach a demultiplexing operation where an incoming packet is placed into a buffer. Carrozza however teaches the above-mentioned limitation in figure 5 and 6 for memory allocation of data demultiplexing (see column 12, lines 56-57).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use the unmapping and mapping of buffers taught by Bugnion with the demultiplexing process taught by Carrozza. The motivation to do so would be to support a buffer that is shared by multiple virtual machines (see the conclusion paragraph in column 7 in Bugnion).

In further regards to claim 12, Bugnion and Carrozza fail to teach allocating unmapped buffers to the virtual machine to create a mapped buffer. Wang however, teaches the above-mentioned limitation. Wang teaches than an API function allocates to a process the physical memory pages that may be mapped and unmapped within any specially-allocated virtual address space region of the specified process (see column 7, lines 10-13).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the buffer allocation taught by Wang into the unmapping and mapping of buffers taught by Bugnion and the demultiplexing process taught by Carrozza. The motivation to do so would be to allow for fast mapping for a multiprocessor system (see column 2, lines 9-12).

In further regards to claim 12, Bugnion, Carrozza and Wang fail to teach the concept of unmapping a buffer space from a first virtual machine and the mapping the same space to a second virtual machine. Paladini however teaches the above-mentioned concept. Paladini teaches un-mapping a buffer from the virtual address space of the application (un mapping from a first virtual machine) and mapping it back later to a different virtual address (mapping it to a second virtual machine) different from the initial assigned address (see paragraph 58 on page 4).

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to incorporate the concept of unmapping and mapping a buffer from one machine to another as taught by Paladini into the teachings of Bugnion, Carrozza and Wang. The motivation to do so would be to use an updated memory manager.

In regards to claim 13, since Bugnion teaches that the transparent migration requires that all mappings that point to a page be removed form all processors (see column 14, lines 19-30), it also reads on clearing the contents of a physical page associated with the host physical address.

In regards to claim 14, the mechanism shown in figure 3 of Bugion, allows for a support of system-wide cache in memory that can be shared between all virtual machines (therefore, reading on a temporary association between a mapped and an unmapped buffer).

In regards to claim 15, since Bugnion teaches that the transparent migration requires that all mappings that point to a page be removed form all processors (see column 14, lines 19-30), it also reads on causing the VM to release the mapped buffer and unmapping the guest physical address from the host physical address.

In regards to claims 16 and18, disco intercepts (injecting a signal and intercepting) all device accesses from the virtual machines and forwards them to the physical address (see column 14, lines 32-34 in Bugnion).

In regards to claim 17, disco uses inter-processor interrupts for specific actions that change the state of a remote virtual processor (see column 10, lines 47-51 in Buginion).

7. In regards to claim 19, Bugnion shows in figure 3, a memory management data structure consisting of two virtual machines with two examples of memory management. The second example shows the impact of a page migration action. Also evident from the figure itself, a virtual address is indexed to a physical address. The transparent migration requires that all mappings that point to that page be removed from all processors (decoupling a guest physical address for a virtual machine from a host physical address to create unmapped buffers) (see column 14, lines 19-30).

In further regards to claim 19, Bugnion fails to teach a demultiplexing operation where an incoming packet is placed into a buffer. Carrozza however teaches the above-mentioned limitation in figure 5 and 6 for memory allocation of data demultiplexing (see column 12, lines 56-57). Furthermore, each incoming packet has a destination address therefore, Carrozza also reads on examining the incoming packets to determine appropriate destination VMs.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to use the unmapping and mapping of buffers taught by Bugnion with the demultiplexing process taught by Carrozza. The motivation to do so would be to support a buffer that is shared by multiple virtual machines (see the conclusion paragraph in column 7 in Bugnion).

In further regards to claim 19, Bugnion and Carrozza fail to teach allocating unmapped buffers to the virtual machine to create a mapped buffer. Wang however, teaches the above-mentioned limitation. Wang teaches than an API function allocates to a process the physical memory pages that may be mapped and unmapped within any specially-allocated virtual address space region of the specified process (see column 7, lines 10-13).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the buffer allocation taught by Wang into the unmapping and mapping of buffers taught by Bugnion and the demultiplexing process taught by Carrozza. The motivation to do so would be to allow for fast mapping for a multiprocessor system (see column 2, lines 9-12).

In further regards to claim 19, Bugnion, Carrozza and Wang fail to teach the concept of unmapping a buffer space from a first virtual machine and the mapping the same space to a second virtual machine. Paladini however teaches the above-mentioned concept. Paladini teaches un-mapping a buffer from the virtual address space of the application (un mapping from a first virtual machine) and mapping it back later to a different virtual address (mapping it to a second virtual machine) different from the initial assigned address (see paragraph 58 on page 4).

Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to incorporate the concept of unmapping and mapping a buffer from one machine to another as taught by Paladini into the teachings of Bugnion, Carrozza and Wang. The motivation to do so would be to use an updated memory manager.

In regards to claim 20, since Bugnion teaches that the transparent migration requires that all mappings that point to a page be removed form all processors (see column 14, lines 19-30), it also reads on invalidating entries in at least one page in a DMA table (also see column 14, lines 28-30 and figure 3).

### ***Response to Arguments***

Applicant's arguments with respect to claim 1 have been considered but are moot in view of the new ground(s) of rejection.

### ***Conclusion***

Any inquiry concerning this communication or earlier communications from the examiner should be directed to JAY P. PATEL whose telephone number is (571)272-3086. The examiner can normally be reached on Mon.-Thurs.: 8:00 a.m.- 6:30 p.m..

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Daniel J. Ryman can be reached on (571)272-3152. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/J. P. P./
Examiner, Art Unit 2419

/DANG T TON/

Supervisory Patent Examiner, Art Unit 2419/D. T. T./

Supervisory Patent Examiner, Art Unit 2419